
프로그래밍 언어 응용

정기 수행 평가 02



Contents

1 라이브러리 설정

- connector lib 적용

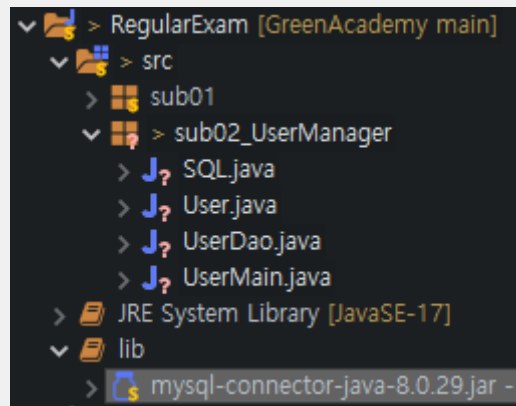
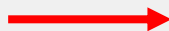
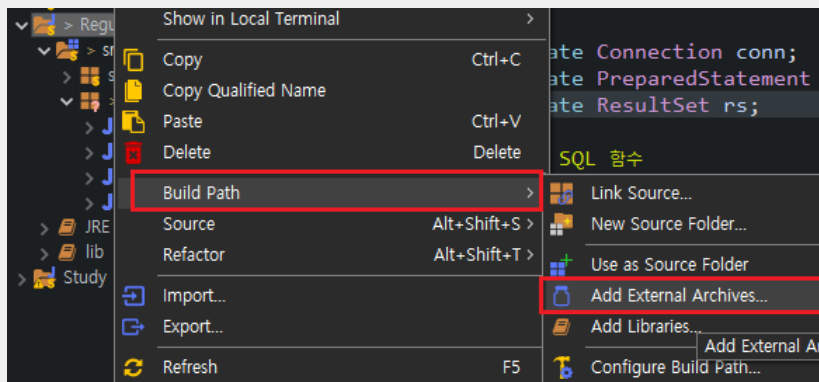
2 프로그램 구성

- SQL 생성
- User 생성
- UserDao 생성
- UserMain 생성

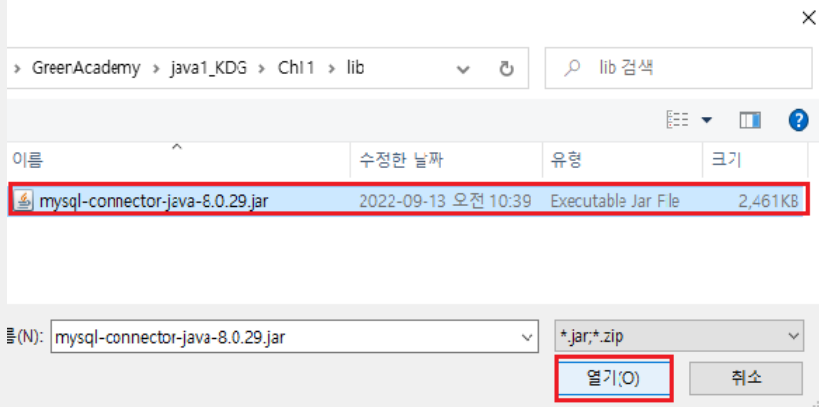
3 실행 화면

1 라이브러리 설정

| connector lib 적용



라이브러리 추가 확인



2 프로그램 구성

| SQL 생성

SQL 구문을 담은 클래스 생성

PreparedStatement을 이용하는 구문 저장

```
1 package sub02_UserManager;
2
3 public class SQL {
4     public static final String INSERT_USER = "INSERT INTO `User3` VALUES(?,?,?,?);";
5     public static final String SELECT_USERS = "SELECT * FROM `User3`;";
6     public static final String SELECT_USER = "SELECT * FROM `User3` WHERE `name`=?";
7     public static final String DELETE_USER = "DELETE FROM `User3` WHERE `uid`=?";
8 }
```

2 프로그램 구성

| User 생성

```
1 package sub02_UserManager;
2
3 public class User {
4     private String uid;
5     private String name;
6     private String hp;
7     private int age;
8
9     //get set
10    public String getUid() {
13    public void setUid(String uid) {
16    public String getName() {
19    public void setName(String name) {
22    public String getHp() {
25    public void setHp(String hp) {
28    public int getAge() {
31    public void setAge(int age) {
34    //get set-end
35
36    @Override
37    public String toString() {
38        return uid + "," + name + "," + hp + "," + age;
39    }
40
41
42 }
```

가져온 데이터를 담은 User 클래스 생성

멤버 접근을 위한 Getter, Setter와

정보 표시를 위한 toString메서드 Overriding

2 프로그램 구성

| UserDao 생성 - 1

```
1 package sub02_UserManager;
2
3 import java.sql.Connection;
4 import java.sql.DriverManager;
5 import java.sql.PreparedStatement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8 import java.sql.SQLIntegrityConstraintViolationException;
9 import java.util.ArrayList;
10 import java.util.List;
11
12 public class UserDao {
13     //싱글톤
14     private static UserDao instance = new UserDao();
15     public static UserDao getInstance() {
16         return instance;
17     }
18     private UserDao() {}
19
20     //DB정보
21     String host = "jdbc:mysql://127.0.0.1:3306/java1db";
22     private String user = "root";
23     private String pass= "1234";
24
25     private Connection getConnection() throws ClassNotFoundException, SQLException{
26         Class.forName("com.mysql.cj.jdbc.Driver");
27         Connection conn = DriverManager.getConnection(host,user,pass);
28         return conn;
29     }
30
31     private Connection conn;
32     private PreparedStatement pstmt;
33     private ResultSet rs;
```

실질적으로 SQL 구문을 전달하는
UserDao 클래스 생성

싱글톤 객체로 생성

DB연결을 위한 getConnection 함수 정의

SQL 구문에 쓰일 멤버 변수들 정의

2 프로그램 구성

| UserDao 생성 - 2

```
//DB SQL 함수
//insertUser
public void insertUser(User user) {
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(SQL.INSERT_USER);
        pstmt.setString(1, user.getId());
        pstmt.setString(2, user.getName());
        pstmt.setString(3, user.getHp());
        pstmt.setInt(4, user.getAge());

        pstmt.executeUpdate();

        close();
    } catch (SQLIntegrityConstraintViolationException e) {
        System.out.println("아이디 또는 휴대폰 중복입니다.");
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }
} //insertUser-end
```

```
//selectUsers
public List<User> selectUsers(){
    List<User> users = new ArrayList<>();

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(SQL.SELECT_USERS);
        rs = pstmt.executeQuery();

        while(rs.next()) {
            User user = new User();
            user.setId(rs.getString(1));
            user.setName(rs.getString(2));
            user.setHp(rs.getString(3));
            user.setAge(rs.getInt(4));

            users.add(user);
        }
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }

    return users;
} //selectUsers-end
```

SQL 구문 전달을 위한 함수

2 프로그램 구성

| UserDao 생성 - 3

```
//selectUser
public User selectUser(String name){

    User user = new User();

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(SQL.SELECT_USER);
        pstmt.setString(1, name);

        rs = pstmt.executeQuery();

        if(rs.next()) {
            user.setUid(rs.getString(1));
            user.setName(rs.getString(2));
            user.setHp(rs.getString(3));
            user.setAge(rs.getInt(4));
        }
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }

    return user;
}
//selectUser-end
```

```
//DeleteUser
public int deleteUser(String name) {

    int result = 0;

    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(SQL.DELETE_USER);
        pstmt.setString(1, name);

        result = pstmt.executeUpdate();

        close();

    } catch (Exception e) {
        e.printStackTrace();
    }

    return result;
} //DeleteUser-end

//closeJDBC
private void close() throws SQLException {
    if(rs != null) rs.close();
    if(pstmt != null) pstmt.close();
    if(conn != null) conn.close();
}
//closeJDBC-end
```


2 프로그램 구성

| UserMain 생성

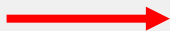
```
1 package sub02_UserManager;
2
3 import java.util.List;
4 import java.util.Scanner;
5
6 public class UserMain {
7
8     public static void main(String[] args) {
9
10         System.out.println("*****");
11         System.out.println("회원관리매니저 v1.0");
12         System.out.println("*****");
13
14         Scanner sc = new Scanner(System.in);
15
16         while(true) {
17             System.out.println("-----");
18             System.out.println("종료:0, 입력:1, 조회:2, 검색:3, 삭제:4");
19             System.out.print("선택 : ");
20             int answer = sc.nextInt();
21
22             if(answer==0) break;
23             else if (answer == 1) {
24                 //데이터 입력
25                 User user = new User();
26
27                 System.out.print("아이디 입력 : ");
28                 user.setUid(sc.next());
29                 System.out.print("이름 입력 : ");
30                 user.setName(sc.next());
31                 System.out.print("휴대폰 입력 : ");
32                 user.setHp(sc.next());
33                 System.out.print("나이 입력 : ");
34                 user.setAge(sc.nextInt());
35
36                 UserDao.getInstance().insertUser(user);
37                 System.out.println("입력완료...");
38             } else if (answer == 2) {
39                 //데이터 조회
40                 List<User> users = UserDao.getInstance().selectUsers();
41
42                 System.out.println("-----전체조회-----");
43                 for(User user : users) {
44                     System.out.println(user);
45                 }
46             } else if (answer == 3) {
47                 //데이터 검색
48                 System.out.print("이름검색 : ");
49                 String name = sc.next();
50
51                 User user = UserDao.getInstance().selectUser(name);
52
53                 System.out.println("-----검색결과-----");
54                 System.out.println(user);
55             } else if (answer == 4) {
56                 //데이터 삭제
57                 System.out.print("삭제아이디 : ");
58                 String uid = sc.next();
59
60                 int result = UserDao.getInstance().deleteUser(uid);
61
62                 if(result >= 1) {
63                     System.out.println("아이디" + uid + "삭제완료");
64                 } else {
65                     System.out.println("아이디" + uid + "사용자가 존재하지 않습니다.");
66                 }
67             }
68         } //while-end
69
70         sc.close();
71         System.out.println("프로그램 종료....");
72     } //main-end
73 }
74
75
76 }
```

메인 프로그램 구현

3 실행화면

| 콘솔 화면 1

```
*****
회원관리매니저 v1.0
*****
-----
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 1
```



```
*****
회원관리매니저 v1.0
*****
-----
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 1
아이디 입력 : test
이름 입력 : test
휴대폰 입력 : 010-test-test
나이 입력 : 27
입력완료...
```

```
-----
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 2
```

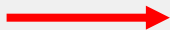


```
-----
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4
선택 : 2
-----전체조회-----
a101,김동근,010-1111-1111,23
a102,김유신,010-2222-2222,120
a103,김동근,010-1111-1113,23
test,test,010-test-test,27
-----
```

3 실행화면

| 콘솔 화면 2

```
-----  
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4  
선택 : 3
```



```
-----  
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4  
선택 : 3  
이름검색 : 김동근  
-----검색결과-----  
a101,김동근,010-1111-1111,23  
-----
```

```
-----  
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4  
선택 : 4
```



```
-----  
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4  
선택 : 4  
삭제아이디 : test  
아이디test삭제완료  
-----
```



```
-----검색결과-----  
a101,김동근,010-1111-1111,23  
-----  
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4  
선택 : 4  
삭제아이디 : test  
아이디test삭제완료  
-----  
종료:0, 입력:1, 조회:2, 검색:3, 삭제:4  
선택 : 2  
-----전체조회-----  
a101,김동근,010-1111-1111,23  
a102,김유신,010-2222-2222,120  
a103,김동근,010-1111-1113,23  
-----
```



정기 수행 평가

02.프로그래밍 언어 응용
UserManager

Thank you!

Kim Dong Geun
From GreenAcademy
2022.09.15